

YAS.021

Record (See the Reference Citation List to obtain the  
citation)

Claims 1-4

Citation 1

Remarks:

In Citation 1, reference is made (reference Sections 0018 to 0020) to an invention for task allocation (corresponding to a "task processor") for each type of event (corresponding to a "storage device which stores the classification elements of an event" as referred to in the present application), in which the same events are processed in the same element task group (corresponding to "further processing of the same task as a task for which processing has been completed") (Reference Sections 0018 to 0020). Furthermore, it is clear that task resources are released when processing is completed (reference is made to the essence of waiting for the release of an element task).

Furthermore, making the invention relating to Claims 1-4 of the present application could be easily conceived by one skilled in the Art on the basis of the notation of Citation 1.

Reference Citation List

1. Japanese Laid Open Patent Publication Hei 8-314739

[First Hit](#)

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)



Generate Collection

Print

L1: Entry 1 of 2

File: JPAB

Nov 29, 1996

PUB-NO: JP408314739A

DOCUMENT-IDENTIFIER: JP 08314739 A

TITLE: MULTITASK CONTROL METHOD AND SYSTEM THEREFOR

PUBN-DATE: November 29, 1996

INVENTOR-INFORMATION:

NAME

COUNTRY

MAEDA, TAMOTSU

ASSIGNEE-INFORMATION:

NAME

COUNTRY

NEC CORP

APPL-NO: JP07119613

APPL-DATE: May 18, 1995

INT-CL (IPC): G06 F 9/46; G06 F 15/16

ABSTRACT:

PURPOSE: To prevent deterioration of the overall throughput of a system by allocating effectively a task to the processing request of any type of resources to avoid occurrence of a task waiting state for a job of the multitask constitution.

CONSTITUTION: An event 102 that is reported from another job is once stored in an event store queue 107 and then taken out. A proper task 105 is allocated to the event 102 when the idle task 105 exists in a group including a slave task that applies the processing to the event 102. If such task 105 does not exist and a shared task 106 exists, the task 106 is allocated to the event 102. The event 102 waits for allocation of a due slave task if the slave task to be allocated to the event 102 does not exist.

COPYRIGHT: (C)1996,JPO

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

JDS # 8  
10/29/04

[First Hit](#)      [Previous Doc](#)      [Next Doc](#)      [Go to Doc#](#)

End of Result Set

☐ [Generate Collection](#) [Print](#)

L1: Entry 2 of 2

File: DWPI

Nov 29, 1996

DERWENT-ACC-NO: 1997-069516

DERWENT-WEEK: 199707

COPYRIGHT 2005 DERWENT INFORMATION LTD

TITLE: Multitasking control method using C system to control number of tasks assigned according to load or resources used by each task - involves assigning share task to received event when share task is not used, and assigning peculiar task to received event when peculiar task in task gp. corresp. to received event is not used

PATENT-ASSIGNEE:

ASSIGNEE

CODE

NEC CORP

NIDE

PRIORITY-DATA: 1995JP-0119613 (May 18, 1995)

[Search Selected](#)

[Search ALL](#)

[Clear](#)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
<input type="checkbox"/> <a href="#">JP 08314739 A</a>	November 29, 1996		011	G06F009/46

APPLICATION-DATA:

PUB-NO	APPL-DATE	APPL-NO	DESCRIPTOR
JP 08314739A	May 18, 1995	1995JP-0119613	

INT-CL (IPC): [G06 F 9/46](#); [G06 F 15/16](#)

ABSTRACTED-PUB-NO: JP 08314739A

BASIC-ABSTRACT:

The method involves simultaneous execution of sub-tasks in the subordinate of a main task (104). A peculiar task (105) is accessed in a task gp. which is collected into a resource unit. The peculiar task is a sub-task which can access only specific resources. A share task (106) is a sub-task which can access all resources.

The main task receives an event (103) by which notification from other jobs is executed. When the peculiar task in the task gp. corresp. to the received event is not used, the peculiar task is assigned to the received event. When the share task is not used, the share task is assigned to the received event.

USE/ADVANTAGE - For e.g. virtual computer system. Enables efficient assigning of task, and prevents waiting for task, thus redn. of system throughput is prevented.

CHOSEN-DRAWING: Dwg.1/6

TITLE-TERMS: CONTROL METHOD SYSTEM CONTROL NUMBER TASK ASSIGN ACCORD LOAD RESOURCE  
TASK ASSIGN SHARE TASK RECEIVE EVENT SHARE TASK ASSIGN PECULIAR TASK RECEIVE EVENT  
PECULIAR TASK TASK GROUP CORRESPOND RECEIVE EVENT

DERWENT-CLASS: T01

EPI-CODES: T01-F02;

SECONDARY-ACC-NO:

Non-CPI Secondary Accession Numbers: N1997-057318

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-314739

(43) 公開日 平成8年(1996)11月29日

(51) Int. Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/46	3 4 0		G 0 6 F 9/46	3 4 0 F
				3 4 0 B
15/16	3 9 0		15/16	3 9 0 Z

審査請求 有 請求項の数10 O L (全 11 頁)

(21) 出願番号 特願平7-119613

(22) 出願日 平成7年(1995)5月18日

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 真栄田 保

東京都港区芝五丁目7番1号 日本電気株式会社内

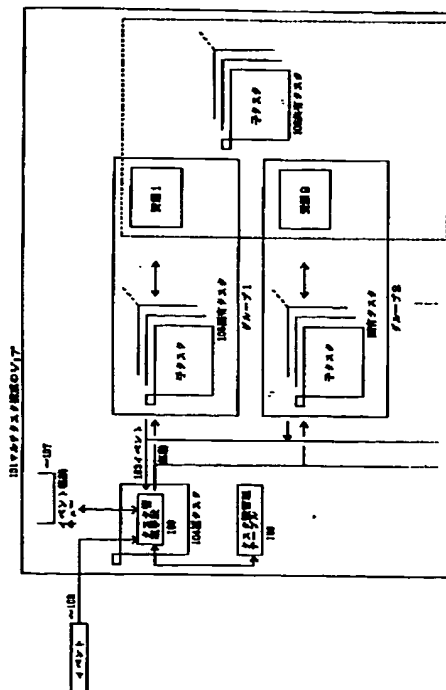
(74) 代理人 弁理士 京本 直樹 (外2名)

(54) 【発明の名称】 マルチタスク制御方法および制御システム

(57) 【要約】

【目的】マルチタスク構成のジョブにおいて、どの資源に対する処理要求であっても、タスクを効率良く割り当て、タスク待ちが発生しないようにすることにより、システム全体のスループットを低下させない。

【構成】他のジョブから通知されたイベント102は一旦イベント格納キュー107に格納された後取り出される。該イベント102に対しては、該イベント102に対する処理を行う子タスクを含むグループに空きの固有タスク105が存在する場合には、該固有タスク105が割り当てられ、空きの固有タスク105が存在しない場合で、かつ空きの共有タスク106が存在する場合には、該共有タスク106が割り当てられる。該イベント102に対して割り当てる子タスクが存在しない場合には、該イベント102は子タスクの割り当て待ちとなる。



1

## 【特許請求の範囲】

【請求項1】 1つの親タスクの配下で複数の子タスクを同時に実行することができる複数のタスクから構成され、該子タスクの内、特定の資源にのみアクセスすることができる子タスクである固有タスクとすべての資源にアクセスすることができる子タスクである共有タスクとをアクセスする資源単位にまとめたタスクグループを備えたジョブにおけるマルチタスク制御方法であって、前記親タスクは、

他のジョブから通知されたイベントを受信し、受信した該イベントに対応する前記タスクグループに未使用の前記固有タスクがある場合には、該固有タスクを該イベントに割り当て、受信した該イベントに対応する前記タスクグループに未使用の前記固有タスク数はないが、未使用の前記共有タスクがある場合には、該共有タスクを該イベントに割り当てることを特徴とするマルチタスク制御方法。

【請求項2】 1つの親タスクの配下で複数の子タスクを同時に実行することができる複数のタスクから構成され、該子タスクの内、特定の資源にのみアクセスすることができる子タスクである固有タスクとすべての資源にアクセスすることができる子タスクである共有タスクとをアクセスする資源単位にまとめたタスクグループと、他のジョブから通知されるイベントを格納するイベント格納キューと、前記タスクグループ毎に設定したエントリが該タスクグループのグループ名、使用可能な固有タスク数、使用中の固有タスク数、使用可能な共有タスク数、および使用中の共有タスク数とを含み、さらに計算機システム内において使用可能な共有タスク数と、計算機システム内において使用中の共有タスク数の合計とを含むタスク数管理テーブルとを備えたジョブにおけるマルチタスク制御方法であって、

前記親タスクは、他のジョブから通知された前記イベントを前記イベント格納キューに格納し、

前記イベントが格納された前記イベント格納キューから該イベントを1つ取り出し、

前記タスク数管理テーブルを参照して、取り出した該イベントに対応する前記タスクグループの前記使用中の固有タスク数が前記使用可能な固有タスク数より小さい場合には、該イベントに前記固有タスクを割り当て、

前記タスク数管理テーブルを参照して、取り出した該イベントに対応する前記タスクグループの前記使用中の固有タスク数が前記使用可能な固有タスク数以上で、かつ前記使用中の共有タスク数が前記使用可能な共有タスク数より小さく、前記計算機システム内において使用中の共有タスク数の合計が前記計算機システム内において使用可能な共有タスク数より小さい場合には、該イベントに前記共有タスクを割り当てることを特徴とするマルチタスク制御方法。

2

【請求項3】 前記タスク数管理テーブルの前記計算機システム内での使用可能共有タスク数、前記使用可能固有タスク数、および前記使用可能共有タスク数は、予め任意の値を設定され、前記使用可能共有タスク数は前記計算機システム内での使用可能共有タスク数以下に、そして各タスクグループの前記使用可能固有タスク数の合計と前記計算機システム内での使用可能共有タスク数とを合算した値は、計算機システム内で起動することができる子タスクの総数以下となるように設定されることを特徴とする請求項2記載のマルチタスク制御方法。

【請求項4】 前記親タスクは、他のジョブから通知されたものであるか前記固有タスクおよび前記共有タスクのいずれかから前記親タスクへ通知されたものであるかを示す種別と、該イベントがいずれの前記タスクグループに対応するものであるかを示すグループ名と、イベントの内容であるイベント情報とを含む前記イベントを受け取った際に、該イベントの前記種別を参照し、該種別が他のジョブから通知されたものであることを示している場合には、該イベントを前記イベント格納キューに格納することを特徴とする請求項2および請求項3記載のマルチタスク制御方法。

【請求項5】 前記親タスクは、前記イベントを受け取った際に、該イベントの前記種別を参照し、該種別が前記固有タスクおよび前記共有タスクのいずれかから前記親タスクへ通知されたものであることを示している場合には、該イベントを通知した該固有タスクおよび該共有タスクのいずれかを解放することを特徴とする請求項4記載のマルチタスク制御方法。

【請求項6】 1つの親タスクの配下で複数の子タスクを同時に実行することができるマルチタスク構成のジョブにおけるマルチタスク制御システムであって、前記ジョブは、

前記子タスクの内、特定の資源にのみアクセスすることができる子タスクである固有タスクとすべての資源にアクセスすることができる子タスクである共有タスクとをアクセスする資源単位にまとめたタスクグループを備え、

前記親タスクは、他のジョブから通知されたイベントを受信し、該イベントに対応する前記タスクグループに未使用の前記固有タスクがある場合には、該固有タスクを該イベントに割り当て、該タスクグループに未使用の前記固有タスク数はないが、未使用の前記共有タスクがある場合には、該共有タスクを該イベントに割り当てることを特徴とするマルチタスク制御システム。

【請求項7】 1つの親タスクの配下で複数の子タスクを同時に実行することができるマルチタスク構成のジョブにおけるマルチタスク制御システムであって、

前記ジョブは、前記子タスクの内、特定の資源にのみアクセスすることができる子タスクである固有タスクとすべての資源にア

アクセスすることができる子タスクである共有タスクとをアクセスする資源単位にまとめたタスクグループと、他のジョブから通知されるイベントを格納するイベント格納キューと、

前記タスクグループ毎に設定したエントリが該タスクグループのグループ名、使用可能な固有タスク数、使用中の固有タスク数、使用可能な共有タスク数、および使用中の共有タスク数とを含み、さらに計算機システム内において使用可能な共有タスク数と、計算機システム内において使用中の共有タスク数の合計とを含むタスク数管理テーブルとを備え、

前記親タスクは、他のジョブから通知された前記イベントを前記イベント格納キューに格納するとともに、該イベント格納キューから該イベントを1つ取り出し、前記タスク数管理テーブルを参照して、該イベントに対応する前記タスクグループの前記使用中の固有タスク数が前記使用可能な固有タスク数より小さい場合には、該イベントに前記固有タスクを割り当て、該イベントに対応する前記タスクグループの前記使用中の固有タスク数が前記使用可能な固有タスク数以上で、かつ前記使用中の共有タスク数が前記使用可能な共有タスク数より小さく、前記計算機システム内において使用中の共有タスク数の合計が前記計算機システム内において使用可能な共有タスク数より小さい場合には、該イベントに前記共有タスクを割り当てることを特徴とするマルチタスク制御システム。

【請求項8】 前記タスク数管理テーブルの前記計算機システム内での使用可能共有タスク数、前記使用可能固有タスク数、および前記使用可能共有タスク数は、予め任意の値を設定され、前記使用可能共有タスク数は前記計算機システム内での使用可能共有タスク数以下に、そして各タスクグループの前記使用可能固有タスク数の合計と前記計算機システム内での使用可能共有タスク数とを合算した値は、計算機システム内で起動することができる子タスクの総数以下となるように設定されることを特徴とする請求項7記載のマルチタスク制御システム。

【請求項9】 前記イベントは、他のジョブから通知されたものであるか前記固有タスクおよび前記共有タスクのいずれかから前記親タスクへ通知されたものであるかを示す種別と、該イベントがいずれの前記タスクグループに対応するものであるかを示すグループ名と、イベントの内容であるイベント情報とを含み、前記親タスクは、前記イベントを受け取った際に、該イベントの前記種別を参照し、該種別が他のジョブから通知されたものであることを示している場合には、該イベントを前記イベント格納キューに格納することを特徴とする請求項7および請求項8記載のマルチタスク制御システム。

【請求項10】 前記親タスクは、前記イベントを受け取った際に、該イベントの前記種別を参照し、該種別が

前記固有タスクおよび前記共有タスクのいずれかから前記親タスクへ通知されたものであることを示している場合には、該イベントを通知した該固有タスクおよび該共有タスクのいずれかを解放することを特徴とする請求項9記載のマルチタスク制御システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、1つの親タスクの配下で複数の子タスクを同時に実行することができるマルチタスク構成のジョブにおいて、各タスクが使用する資源の負荷に応じて割り当てるタスク数を制御するマルチタスク制御方法および制御システムに関する。

【0002】

【従来の技術】特開平4-321136号公報に記載された技術（以下、従来の技術とする）は、計算機システムの仮想空間内でCPU実行権を得てプログラムを実行するスレッドを、外部変数やファイル資源などを共用する単位でグループ化し、1つのタスクにより管理している。各タスクにおいては複数のスレッドにより資源が共用されるが、異なるタスクで動作するスレッド間では資源は共用されない。

【0003】この従来の技術においては、このような構成を採用することにより、同一仮想空間内で複数のタスクを同時に実行できるようにしている。

【0004】

【発明が解決しようとする課題】この従来の技術においては、各タスクにおいて動作可能な最大スレッド数が固定されているので、特定の資源に対する処理要求が集中したときにスレッドが不足して、該タスクにおける処理効率が低下してしまうという問題点がある。

【0005】本発明の目的は、ジョブが使用することができるすべての子タスクを特定の資源にアクセスする固有タスクとすべての資源にアクセスすることができる共有タスクとに分け、さらにアクセスする資源単位に固有タスクと共有タスクとを含むグループを設定し、該グループに対するタスクの割り当て要求に応じて固有タスクと共有タスクとを割り当てるとともに各グループ間で共有タスクを競合して取得/解放することにより、どの資源に対する処理要求であっても、タスクを効率良く割り当て、タスク待ちが発生しないようにすることにある。

【0006】本発明の他の目的は、どの資源に対する処理要求であっても、タスクを効率良く割り当て、タスク待ちが発生しないようにすることにより、システム全体のスループットを低下させないようにすることにある。

【0007】

【課題を解決するための手段】本発明の第1のマルチタスク制御方法は、1つの親タスクの配下で複数の子タスクを同時に実行することができる複数のタスクから構成され、該子タスクの内、特定の資源にのみアクセスすることができる子タスクである固有タスクとすべての資源

5

にアクセスすることができる子タスクである共有タスクとをアクセスする資源単位にまとめたタスクグループを備えたジョブにおけるマルチタスク制御方法であって、前記親タスクは、他のジョブから通知されたイベントを受信し、受信した該イベントに対応する前記タスクグループに未使用の前記固有タスクがある場合には、該固有タスクを該イベントに割り当て、受信した該イベントに対応する前記タスクグループに未使用の前記固有タスク数はないが、未使用の前記共有タスクがある場合には、該共有タスクを該イベントに割り当てることを特徴とする。

【0008】本発明の第2のマルチタスク制御方法は、1つの親タスクの配下で複数の子タスクを同時に実行することができる複数のタスクから構成され、該子タスクの内、特定の資源にのみアクセスすることができる子タスクである固有タスクとすべての資源にアクセスすることができる子タスクである共有タスクとをアクセスする資源単位にまとめたタスクグループと、他のジョブから通知されるイベントを格納するイベント格納キューと、前記タスクグループ毎に設定したエントリが該タスクグループのグループ名、使用可能な固有タスク数、使用中の固有タスク数、使用可能な共有タスク数、および使用中の共有タスク数とを含み、さらに計算機システム内において使用可能な共有タスク数と、計算機システム内において使用中の共有タスク数の合計とを含むタスク数管理テーブルとを備えたジョブにおけるマルチタスク制御方法であって、前記親タスクは、他のジョブから通知された前記イベントを前記イベント格納キューに格納し、前記イベントが格納された前記イベント格納キューから該イベントを1つ取り出し、前記タスク数管理テーブルを参照して、取り出した該イベントに対応する前記タスクグループの前記使用中の固有タスク数が前記使用可能な固有タスク数より小さい場合には、該イベントに前記固有タスクを割り当て、前記タスク数管理テーブルを参照して、取り出した該イベントに対応する前記タスクグループの前記使用中の固有タスク数が前記使用可能な固有タスク数以上で、かつ前記使用中の共有タスク数が前記使用可能な共有タスク数より小さく、前記計算機システム内において使用中の共有タスク数の合計が前記計算機システム内において使用可能な共有タスク数より小さい場合には、該イベントに前記共有タスクを割り当てることを特徴とする。

【0009】本発明の第3のマルチタスク制御方法は、第2のマルチタスク制御方法において、前記タスク数管理テーブルの前記計算機システム内での使用可能共有タスク数、前記使用可能固有タスク数、および前記使用可能共有タスク数は、予め任意の値を設定され、前記使用可能共有タスク数は前記計算機システム内での使用可能共有タスク数以下に、そして各タスクグループの前記使用可能固有タスク数の合計と前記計算機システム内での

6

使用可能共有タスク数とを合算した値は、計算機システム内で起動することができる子タスクの総数以下となるように設定されることを特徴とする。

【0010】本発明の第4のマルチタスク制御方法は、第2および第3のマルチタスク制御方法において、前記親タスクは、他のジョブから通知されたものであるか前記固有タスクおよび前記共有タスクのいずれかから前記親タスクへ通知されたものであるかを示す種別と、該イベントがいずれの前記タスクグループに対応するものであるかを示すグループ名と、イベントの内容であるイベント情報とを含む前記イベントを受け取った際に、該イベントの前記種別を参照し、該種別が他のジョブから通知されたものであることを示している場合には、該イベントを前記イベント格納キューに格納することを特徴とする。

【0011】本発明の第5のマルチタスク制御方法は、第4のマルチタスク制御方法において、前記親タスクは、前記イベントを受け取った際に、該イベントの前記種別を参照し、該種別が前記固有タスクおよび前記共有タスクのいずれかから前記親タスクへ通知されたものであることを示している場合には、該イベントを通知した該固有タスクおよび該共有タスクのいずれかを解放することを特徴とする。

【0012】本発明の第6のマルチタスク制御システムは、1つの親タスクの配下で複数の子タスクを同時に実行することができるマルチタスク構成のジョブにおけるマルチタスク制御システムであって、前記ジョブは、前記子タスクの内、特定の資源にのみアクセスすることができる子タスクである固有タスクとすべての資源にアクセスすることができる子タスクである共有タスクとをアクセスする資源単位にまとめたタスクグループを備え、前記親タスクは、他のジョブから通知されたイベントを受信し、該イベントに対応する前記タスクグループに未使用の前記固有タスクがある場合には、該固有タスクを該イベントに割り当て、該タスクグループに未使用の前記固有タスク数はないが、未使用の前記共有タスクがある場合には、該共有タスクを該イベントに割り当てることを特徴とする。

【0013】本発明の第7のマルチタスク制御システムは、1つの親タスクの配下で複数の子タスクを同時に実行することができるマルチタスク構成のジョブにおけるマルチタスク制御システムであって、前記ジョブは、前記子タスクの内、特定の資源にのみアクセスすることができる子タスクである固有タスクとすべての資源にアクセスすることができる子タスクである共有タスクとをアクセスする資源単位にまとめたタスクグループと、他のジョブから通知されるイベントを格納するイベント格納キューと、前記タスクグループ毎に設定したエントリが該タスクグループのグループ名、使用可能な固有タスク数、使用中の固有タスク数、使用可能な共有タスク数、



および使用中の共有タスク数とを含み、さらに計算機システム内において使用可能な共有タスク数と、計算機システム内において使用中の共有タスク数の合計とを含むタスク数管理テーブルとを備え、前記親タスクは、他のジョブから通知された前記イベントを前記イベント格納キューに格納するとともに、該イベント格納キューから該イベントを1つ取り出し、前記タスク数管理テーブルを参照して、該イベントに対応する前記タスクグループの前記使用中の固有タスク数が前記使用可能な固有タスク数より小さい場合には、該イベントに前記固有タスクを割り当て、該イベントに対応する前記タスクグループの前記使用中の固有タスク数が前記使用可能な固有タスク数以上で、かつ前記使用中の共有タスク数が前記使用可能な共有タスク数より小さく、前記計算機システム内において使用中の共有タスク数の合計が前記計算機システム内において使用可能な共有タスク数より小さい場合には、該イベントに前記共有タスクを割り当てることを特徴とする。

【0014】本発明の第8のマルチタスク制御システムは、第7のマルチタスク制御システムにおいて、前記タスク数管理テーブルの前記計算機システム内での使用可能共有タスク数、前記使用可能固有タスク数、および前記使用可能共有タスク数は、予め任意の値を設定され、前記使用可能共有タスク数は前記計算機システム内での使用可能共有タスク数以下に、そして各タスクグループの前記使用可能固有タスク数の合計と前記計算機システム内での使用可能共有タスク数とを合算した値は、計算機システム内で起動することができる子タスクの総数以下となるように設定されることを特徴とする。

【0015】本発明の第9のマルチタスク制御システムは、第7および第8のマルチタスク制御システムにおいて、前記イベントは、他のジョブから通知されたものであるか前記固有タスクおよび前記共有タスクのいずれかから前記親タスクへ通知されたものであるかを示す種別と、該イベントがいずれの前記タスクグループに対応するものであるかを示すグループ名と、イベントの内容であるイベント情報とを含み、前記親タスクは、前記イベントを受け取った際に、該イベントの前記種別を参照し、該種別が他のジョブから通知されたものであることを示している場合には、該イベントを前記イベント格納キューに格納することを特徴とする。

【0016】本発明の第10のマルチタスク制御システムは、第9のマルチタスク制御システムにおいて、前記親タスクは、前記イベントを受け取った際に、該イベントの前記種別を参照し、該種別が前記固有タスクおよび前記共有タスクのいずれかから前記親タスクへ通知されたものであることを示している場合には、該イベントを通知した該固有タスクおよび該共有タスクのいずれかを解放することを特徴とする。

【0017】

【実施例】以下本発明の一実施例について、図を参照しながら詳細に説明する。

【0018】図1を参照すると、本発明の一実施例であるマルチタスク制御方法が適用されるジョブ101は、処理の過程において同一資源をアクセスする複数の子タスクを含む固有タスク105と、処理の過程においてすべての資源を共有してアクセスすることができる複数の子タスクを含む共有タスク106と、固有タスク105および共有タスク106の起動状態を管理するとともに固有タスク105および共有タスク106を起動する親タスク104と、固有タスク105および共有タスク106から親タスク104へ通知されるイベント103と、他のジョブから通知されるイベント102を格納するイベント格納キュー107と、使用中の固有タスク数と共有タスク数、および未使用の固有タスク数と共有タスク数を管理するタスク数管理テーブル108と、親タスク104においてタスク数管理テーブル108を参照し、イベント格納キュー107から取り出したイベントに子タスクを割り当てるタスク管理手段109とから構成されている。

【0019】なお、資源単位に該資源をアクセスする固有タスク105と共有タスク106をまとめてグループと称する。

【0020】図2を参照すると、イベント102およびイベント103は、他のジョブから通知されたイベント102であるか自ジョブ内の固有タスク105および共有タスク106から親タスク104へ通知されたイベント103であることを示す種別21と、該イベントが自ジョブ内のどの子タスクのグループに対するものであるかを示すグループ名22と、イベントの内容であるイベント情報23とを含む。

【0021】図3を参照すると、タスク数管理テーブル108は、子タスクのN個のグループ31毎にエントリ(行)を設定し、各エントリは、使用可能固有タスク数32、使用可能共有タスク数33、使用中固有タスク数34、および使用中共有タスク数35を含む。さらにタスク数管理テーブル108は、計算機システム内での使用可能共有タスク数36、および計算機システム内での使用中共有タスク数の合計37を含む。

【0022】なお、計算機システム内での使用可能共有タスク数36、各グループの使用可能固有タスク数32、および各グループの使用可能共有タスク数33は、予め任意の値を設定する。この各グループの使用可能共有タスク数33は、計算機システム内での使用可能共有タスク数36以下でなければならない、各グループの使用可能固有タスク数32の合計と計算機システム内での使用可能共有タスク数36とを合算した値は、計算機システム内で起動することができる子タスクの数以下でなければならない。

【0023】次に本発明の一実施例の動作について、図

1〜図6を参照して説明する。

【0024】ジョブ101の親タスク104内のタスク管理手段109は、イベントを受け取ると、該イベントの種別21を判定し(ステップ401)、他のジョブから通知されたイベント102であればイベント格納キュー107へ該イベント102を登録する(ステップ402)。

【0025】タスク管理手段109は、ステップ401において、受け取ったイベントがいずれかの子タスクのグループから通知されたイベント103であると判定した場合10 には、該イベント103に含まれるグループ名22を参照し、タスク数管理テーブル108内の該グループ名22に対応するエントリの使用中共有タスク数35の値を参照する(ステップ404)。

【0026】タスク管理手段109は、この使用中共有タスク数35の値が0より大きい場合には、イベントを通知したグループ内で共有タスク106が処理を終了し解放されたものと判断して、使用中共有タスク数35の値から1減算する(ステップ405)とともに、計算機システム内での使用中共有タスク数の合計37からも1 20 減算する(ステップ406)。

【0027】タスク管理手段109は、ステップ404で参照した使用中共有タスク数35の値が0であれば、イベントを通知したグループ内で固有タスク105が処理を終了し解放されたものと判断して、使用中固有タスク数34から1減算する(ステップ407)。

【0028】タスク管理手段109は、ステップ406またはステップ407終了後、ステップ403へ至る。

【0029】タスク管理手段109は、イベント格納キュー107からイベント102を1つ選択する(ステップ403)。このイベント格納キュー107からのイベント102の選択方法は、FIFO、ラウンドロビン、および動作タスク数が最小のグループから通知されたイベントを選択する等を適用することができるが、本発明においてはどの方法を用いても良く、本実施例においては特定しない。

【0030】タスク管理手段109は、イベント格納キュー107にイベント102が格納されているかどうかを判定し(ステップ408)、格納されていなければイベント102の入力を待ち合わせる(ステップ40 40 9)。

【0031】タスク管理手段109は、ステップ408においてイベント格納キュー107にイベント102が格納されていると判定した場合には、該イベント102を1つ取り出し、該イベント102に含まれるグループ名22を参照する。

【0032】タスク管理手段109は、タスク数管理テーブル108内の該グループ名22に対応するエントリの使用中固有タスク数34と使用可能固有タスク数32を比較し(ステップ410)、使用中固有タスク数34 50

が使用可能固有タスク数32より小さい場合には、該イベント102に固有タスク105を割り当てることができると判断して、使用中固有タスク数34に1加算し(ステップ411)、該イベント102に固有タスク105を割り当てて起動する(ステップ416)。

【0033】タスク管理手段109は、ステップ410において使用中固有タスク数34が使用可能固有タスク数32以上であると判定した場合には、該イベント102に固有タスク105を割り当てることができないと判断して、使用中共有タスク数35と使用可能共有タスク数33とを比較し(ステップ412)、使用中共有タスク数35が使用可能共有タスク数33より小さい場合には、さらに計算機システム内での使用中共有タスク数の合計37と計算機システム内での使用可能共有タスク数36とを比較する(ステップ413)。

【0034】タスク管理手段109は、計算機システム内での使用中共有タスク数の合計37が計算機システム内での使用可能共有タスク数36より小さいと判定した場合には、該イベント102に共有タスク106を割り 10 当てることができると判断して、使用中共有タスク数35に1加算する(ステップ414)とともに、計算機システム内での使用中共有タスク数の合計37にも1加算し(ステップ415)、該イベント102に共有タスク106を割り当てて起動する(ステップ416)。

【0035】タスク管理手段109は、ステップ412において使用中共有タスク数35が使用可能共有タスク数33以上であると判定した場合には、子タスクが解放されるのを待ち合わせる(ステップ417)。

【0036】タスク管理手段109は、ステップ413において計算機システム内での使用中共有タスク数の合計37が計算機システム内での使用可能共有タスク数36以上であると判定した場合には、子タスクが解放されるのを待ち合わせる(ステップ418)。

【0037】タスク管理手段109は、ステップ416において共有タスク106を起動した後は、イベント入力待ちとなる(ステップ419)。

【0038】図5を参照すると、データベースAをアクセスするタスクのグループAと、データベースBをアクセスするタスクのグループBが設定されている。

【0039】図6は、図5のように設定されたタスクのグループを管理するタスク数管理テーブル108の例を示している。

【0040】グループA内のタスクを起動するためのイベントが入力されると、使用可能固有タスク数32が3、使用中固有タスク数34が2であることから、該イベントに固有タスク105を割り当てて起動することができる。

【0041】またグループB内のタスクを起動するためのイベントが入力されると、使用可能固有タスク数32が1、使用中固有タスク数34が1であることから、該

11

イベントに固有タスク105を割り当てることができないため、該イベントに共有タスク106を割り当てて起動することができるかどうかを判定する。このグループBの使用可能共有タスク数33が1、使用中共有タスク数35が0、計算機システム内での使用中共有タスク数の合計37が0であることから共有タスク106が空いているので、該イベントに共有タスク106を割り当てて起動することができる。

【0042】このように本発明は、グループAのタスクの動作に影響することなく、グループBの処理量増大に対処することができ、タスクのグループ間の負荷バランスを制御することができる効果を有している。

【0043】

【発明の効果】以上説明したように、本発明のマルチタスク制御方法および制御システムによれば、ジョブが使用することができるすべてのタスクを特定の資源にのみアクセスすることができる固有タスクとすべての資源にアクセスすることができる共有タスクとに分け、さらにアクセスする資源単位に固有タスクと共有タスクとを含むグループを設定し、該グループへのタスクの割り当て要求に応じて固有タスクと共有タスクとを割り当てるとともに各グループ間で共有タスクを競合して取得／解放することにより、どの資源に対する処理要求であっても、タスクを効率良く割り当て、タスク待ちが発生しないようにすることができる効果を有している。

【0044】また、どの資源に対する処理要求であっても、タスクを効率良く割り当て、タスク待ちが発生しないようにすることにより、システム全体のスループットを低下させないという効果を有している。

【図面の簡単な説明】

12

【図1】図1は本発明の一実施例であるマルチタスク制御方法を適用したジョブの構成を示す図である。

【図2】図2は本発明の一実施例におけるイベント102およびイベント103の内容を示す図である。

【図3】図3は本発明の一実施例におけるタスク数管理テーブル108の内容を示す図である。

【図4】図4は本発明の一実施例におけるタスク管理手段109の処理を示す流れ図である。

【図5】図5は本発明のマルチタスク制御方法を適用したジョブの構成の例を示す図である。

【図6】図6は本発明のマルチタスク制御方法を適用したジョブ内のタスク数管理テーブル108の内容を示す図である。

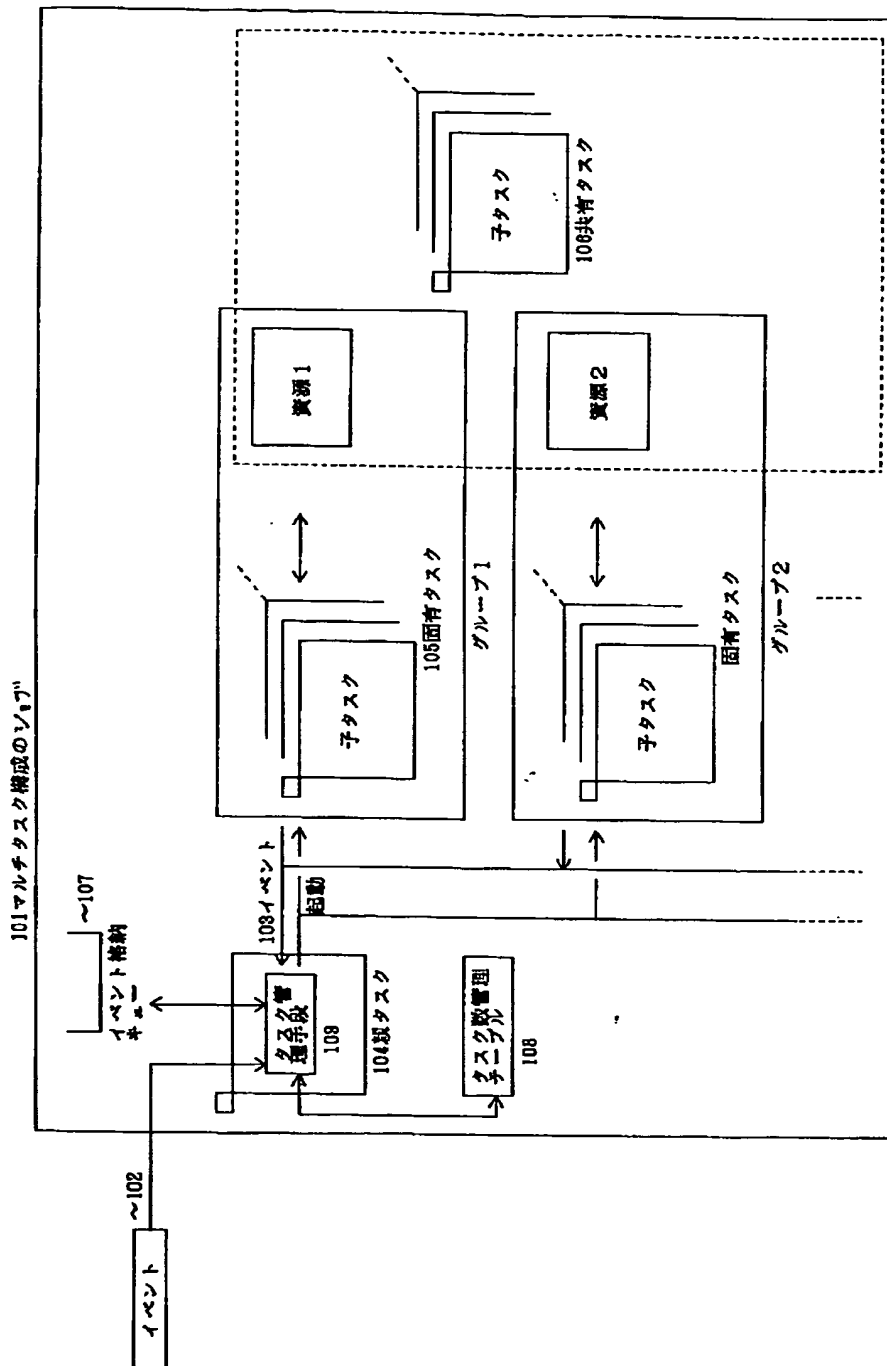
【符号の説明】

- 31 グループ名
- 32 使用可能固有タスク数
- 33 使用可能共有タスク数
- 34 使用中固有タスク数
- 35 使用中共有タスク数
- 36 システム内での使用可能共有タスク数
- 37 システム内の使用中共有タスク数の合計
- 101 ジョブ
- 102 イベント
- 103 イベント
- 104 親タスク
- 105 固有タスク
- 106 共有タスク
- 107 イベント格納キュー
- 108 タスク数管理テーブル
- 109 タスク管理手段

【図2】

21 種別	22 グループ名	23 イベント情報
-------	----------	-----------

【図1】



【図3】

86 システム内での使用可能共有タスク数	
87 システム内での使用中共有タスク数の合計	

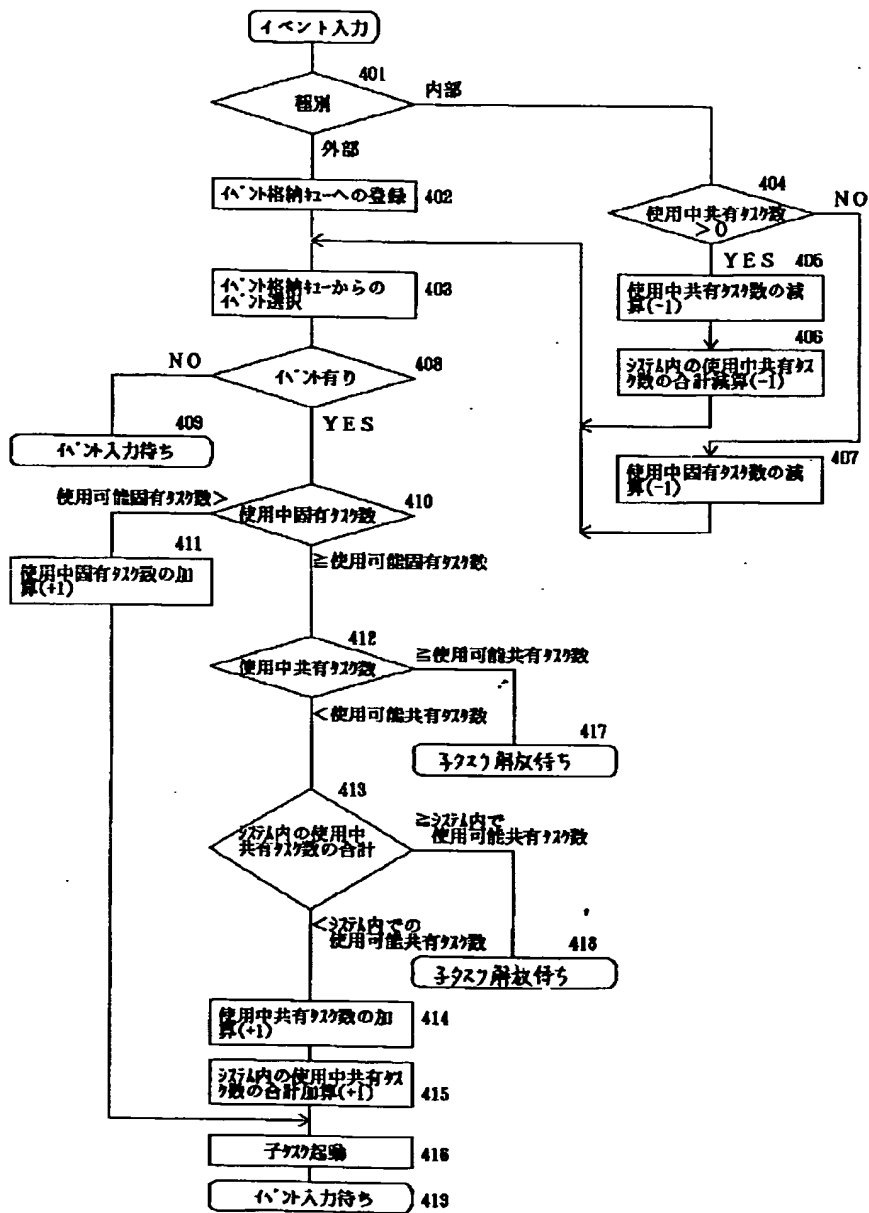
31 グループ名	32使用可能固有タスク数	33使用可能共有タスク数	34使用中固有タスク数	35使用中共有タスク数
1				
.....				
N				

【図6】

システム内での使用可能共有タスク数	3
システム内での使用中共有タスク数の合計	0

資源名	使用可能固有タスク数	使用可能共有タスク数	使用中固有タスク数	使用中共有タスク数
A	3	2	2	0
B	1	1	1	0

【図4】



【図5】

